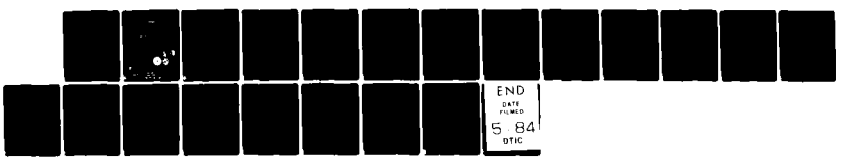
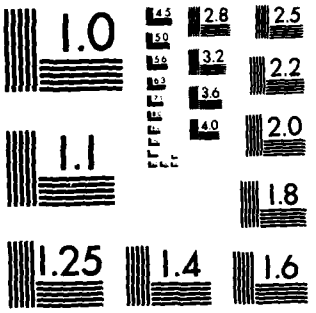
AD-A140 097 BANDED PRECONDITIONING FOR THE SOLUTION OF SYMMETRIC POSITIVE DEFINITE LI..(U) CALIFORNIA UNIV BERKELEY CENTER FOR PURE AND APPLIED MATHEMAT.. 1/1

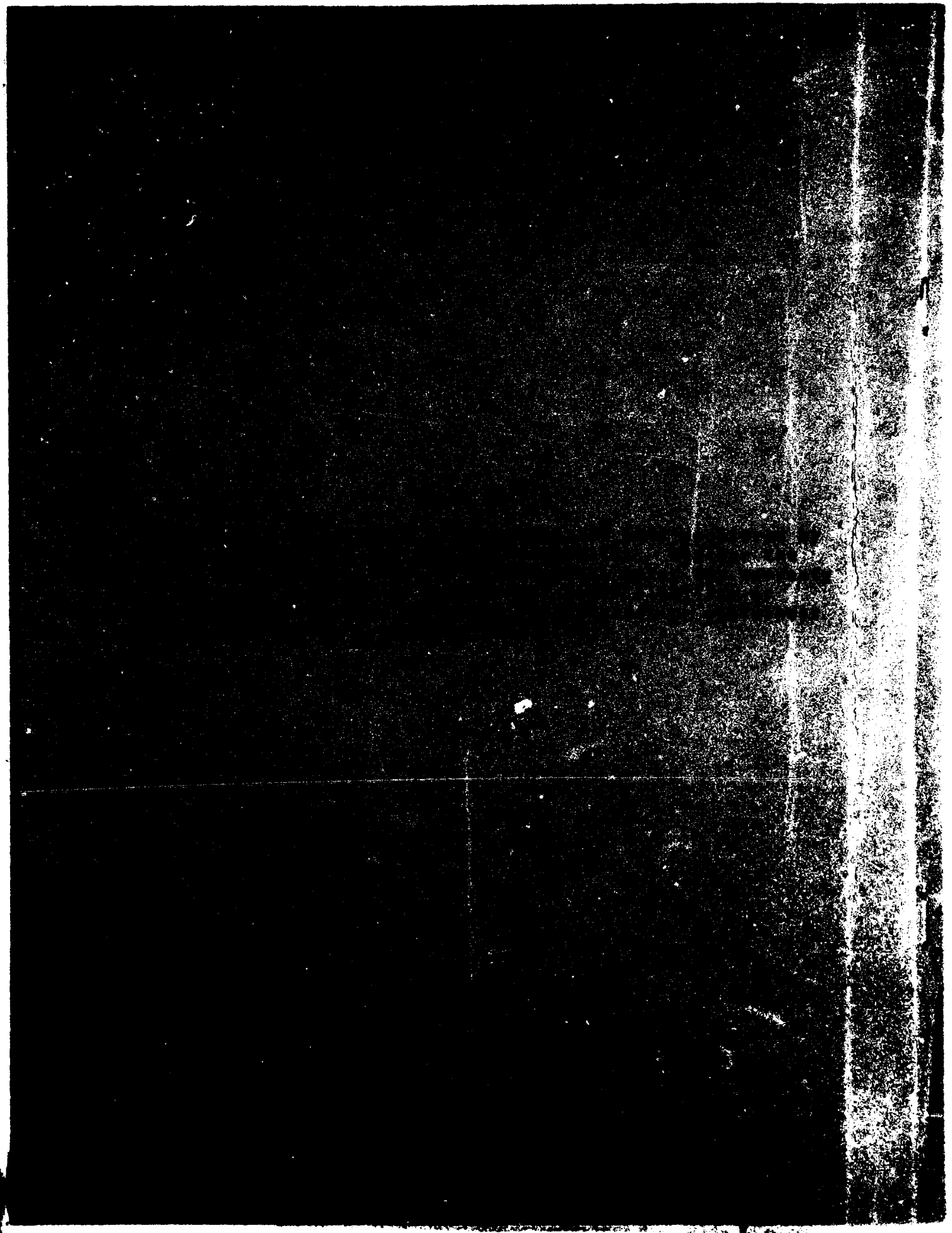
UNCLASSIFIED B NOUR-OMID ET AL. OCT 83 CPAM-185 F/G 12/1 NL



END
DATE FILMED
5 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 185	2. GOVT ACCESSION NO. A140 094	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Banded Preconditioning for the Solution of Symmetric Positive Definite Linear Systems		5. TYPE OF REPORT & PERIOD COVERED Unclassified
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bahram Nour-Omid and Horst D. Simon		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0013
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Berkeley, CA 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE October 1983
		13. NUMBER OF PAGES 8
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The preconditioned conjugate gradient algorithm has been successfully applied to solving symmetric linear systems of equations arising from finite difference and finite element discitizations of a variety of problems.

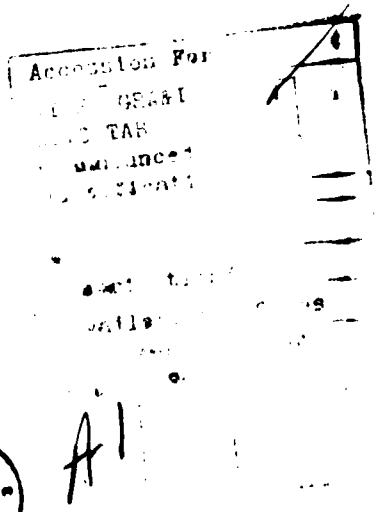
In this paper we consider a matrix splitting, $A = M + R$, where M is a part of A chosen such that its factorization has little or nor fill-in.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

We develop a simple criterion to check for the positive definiteness of M. It turns out that a large class of matrices, including matrices arising from finite element discretization of elliptic boundary value problems, satisfy the criterion.

We present some numerical tests where M is used as a preconditioning matrix for the conjugate gradient algorith. Our examples include problems arising from structural engineering. A comparison with the preconditioning based on an incomplete Choleski factorization is encouraging.



BANDED PRECONDITIONING FOR THE SOLUTION OF SYMMETRIC POSITIVE DEFINITE LINEAR SYSTEMS

by

Bahram Nour-Omid †

and

Horst D. Simon ‡

ABSTRACT

The preconditioned conjugate gradient algorithm has been successfully applied to solving symmetric linear systems of equations arising from finite difference and finite element discitizations of a variety of problems.

In this paper we consider a matrix splitting, $\mathbf{A} = \mathbf{M} + \mathbf{R}$, where $\mathbf{M}$ is a part of $\mathbf{A}$ chosen such that its factorization has little or no fill-in. We develop a simple criterion to check for the positive definiteness of $\mathbf{M}$. It turns out that a large class of matrices, including matrices arising from finite element discretization of elliptic boundary value problems, satisfy this criterion.

We present some numerical tests where $\mathbf{M}$ is used as a preconditioning matrix for the conjugate gradient algorithm. Our examples include problems arising from structural engineering. A comparison with the preconditioning based on an incomplete Choleski factorization is encouraging.

---

† Center for Pure and Applied Mathematics, University of California, Berkeley CA 94563. The research of this author was supported in part by the Office of Naval Research under contract N00014-76-C-0013.

‡ Boeing Computing Services Co., MS 9C-01, Tukwila WA 98188.

1. INTRODUCTION

Exploiting the sparsity structure of large symmetric positive definite matrix **A** when solving linear systems

$$\mathbf{Ax} = \mathbf{b} \tag{1}$$

has become an essential part of acceptable solution techniques. Here we are mainly interested in matrices arising from finite element applications.

The conjugate gradient method (CG hereafter) is one of the more popular iterative methods for solving (1). At each iteration step the CG method requires the computation of $\mathbf{Av}$ for a given vector $\mathbf{v}$. This is the only connection between **A** and the algorithm and is a natural way of making use of the sparsity of **A**.

Past experience has shown that the application of the CG method directly to (1) may require many steps to reduce the residual norm to an acceptable level. This difficulty can be overcome through preconditioning. Instead of solving (1), we attempt to solve

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b} \tag{2}$$

where $\mathbf{M}^{-1}$ is an approximation to $\mathbf{A}^{-1}$. In order to apply CG to (2), **M** must also be symmetric positive definite.

Preconditionings based on SSOR, fast direct methods, or on the block-diagonal part have been effective in stimulating research in this area (see [2,3,8] for more details). One of the successful preconditionings used in connection with CG is the incomplete Choleski factorization of **A** (ICCG) [9,11,13,15]. In some of these papers truly spectacular numerical results are reported. ICCG type methods proved to be far superior to other iterative methods and demonstrated the true potential of preconditioned CG method.

The implementation of ICCG type methods is not difficult for finite difference matrices [13]. If one forces the incomplete Choleski factors to have the same zero structure as that for **A** then for general matrices, in particular for finite element matrices, the data structure for **A** and the incomplete factors are the same and the same array of pointers can be used for both. The data structure however, becomes more complicated if one accepts or rejects fill-in according to the

relative size of the matrix elements as in [15]. For this algorithm the data structure has to be changed during the process of incomplete factorization. This complication prompted us to take a simpler approach to preconditioning. We partition **A** into the sum of two matrices

$$\mathbf{A} \quad = \quad \mathbf{M} \quad + \quad \mathbf{R} \tag{3}$$

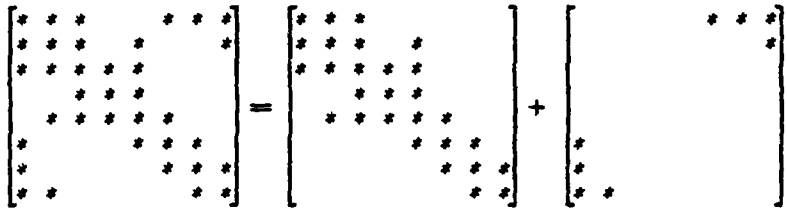


Figure 1. Partitioning of **A** into dense matrix **M** and the remainder **R**.

where **M** is a symmetric part of **A** whose Choleski factors has little or no fill-in (see figure 1). We will refer to **M** as the dense part of **A**. Here we only consider the banded form of **M**, but profile or any other sparse-factorizable form of **M** can also be used. $\mathbf{M}^{-1}$ may now be used as a preconditioner.

Considering the simplicity of this approach, one might ask, why this preconditioning method has not been tried earlier. Recently some similar ideas were reported in [14]. One apparent reason is that **M** (the dense part of **A**) is not positive definite in general. Indeed this restricts the applicability of our method. However we will show that an important class of matrices has a positive definite dense part. A special case of this splitting that is emphasised here is when **M** is banded. We refer to **M** as the banded part of **A** however all the results that follow extend immediately to a more general **M** (profile form etc.).

2. Preconditioned Conjugate Gradients

The initial warm embrace given to conjugate gradient method was due to a number of factors. In exact arithmetic CG cannot take more than n steps to solve $\mathbf{Ax} = \mathbf{b}$ for positive definite **A**. Soon it was found that under certain conditions practical CG may require as many as $3n$ or $4n$ steps to reduce the residual to an acceptable level [4]. This blemish can be accounted for by the strong influence of round-off errors.

The resurrection of CG is due to the addition of preconditioning. The original ill-conditioned problem is transformed into a problem that is better conditioned. There are some theoretical arguments as to why preconditioning is a reasonable technique. Standard results show that the residual norm is reduced by a factor at least $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ at each iteration. κ is the condition number of $\mathbf{A}$, defined as $\kappa(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|$ (see e.g. [1]). When $\mathbf{A}$ is ill-conditioned (i.e. $\kappa(\mathbf{A})$ is large), one can solve equation (2) and choose $\mathbf{M}$ such that the condition of $\mathbf{M}^{-1}\mathbf{A}$ is small. A well chosen $\mathbf{M}$ will require only a few iterations of the CG algorithm to reduce the residual norm to the desired level.

There are two different approaches that one can take in constructing preconditioning matrices:

(i) Incomplete Factorization of the Complete A.

The incomplete Choleski factorization can be viewed this way. Here the Choleski factors of a positive definite matrix $\overline{\mathbf{A}}$, is formed. $\overline{\mathbf{A}}$ is a matrix close to $\mathbf{A}$ with its Choleski factors having the same non-zero structure as the lower triangular part of $\mathbf{A}$. It should be noted that, in general, $\overline{\mathbf{A}}$ is denser than $\mathbf{A}$ and the elements of $\overline{\mathbf{A}}$ are different from the elements of $\mathbf{A}$.

(ii) Complete Factorization of an Incomplete A.

Here one removes those off-diagonal elements of $\mathbf{A}$ that are responsible for fill-in when $\mathbf{A}$ is factored. An example such of approach is diagonal scaling.

These points of view are merely helpful distinctions of preconditioning methods. They do not yield two separate classes of preconditioning matrices as the following example shows: let

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 \\ 0 & -1 & 3 & -1 \\ 0 & -1 & -1 & 7 \end{bmatrix}.$$

Then one obtains the same preconditioning matrix $\mathbf{M}$, whether one uses an incomplete factorization with the factors having a bi-diagonal structure or whether the tridiagonal part of $\mathbf{A}$ is used. Our main concern is general methods that are derived using the second approach.

We give an outline of the CG algorithm here. Given an initial guess $\mathbf{x}_0$ and a positive definite matrix $\mathbf{M}$ the preconditioned CG algorithm (PCG hereafter) generates a sequence $\mathbf{x}_k$ of approximations to the solution $\mathbf{x}$ as follows:

(I) set $\mathbf{p}_0 = r_0 = \mathbf{b} - \mathbf{A}\mathbf{x}$

(II) solve $\mathbf{M}\mathbf{z}_0 = \mathbf{r}_0$,

(III) For $k = 0$, step 1 until convergence do

(a) $\alpha_k = (\mathbf{r}_k, \mathbf{z}_k)/(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)$

(b) $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$

(c) $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$

(d) solve $\mathbf{M}\mathbf{z}_{k+1} = \mathbf{r}_{k+1}$

(e) $\beta_k = (\mathbf{r}_{k+1}, \mathbf{z}_{k+1})/(\mathbf{r}_k, \mathbf{z}_k)$

(f) $\mathbf{p}_{k+1} = \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k$

A nice derivation and discussion of the variants of the basic conjugate gradient algorithm can be found in [16]. The preconditioned version of the algorithm is explained in [3].

3. Positive Definiteness of M

The idea of using the complete factorization of an incomplete A has been raised before. Methods that have been constructed by this approach include diagonal scaling, line methods [8], which can be considered as block diagonal preconditioning, and hybrid method [10] which is a blend of a 2×2 block diagonal factorization into CG iteration. The reason this approach has not been generalized to more sophisticated structures, is that an incomplete A is not positive definite in general. Here we give an example to illustrate this fact.

Example

Consider the 3×3 matrices $\mathbf{A}_3 = \begin{bmatrix} 5 & 4 & 1 \\ 4 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and $\mathbf{T}_3 = \begin{bmatrix} 5 & 4 & 0 \\ 4 & 4 & 1 \\ 0 & 1 & 1 \end{bmatrix}$. It is trivial to show that $\det(\mathbf{A}_3) = 3$ and $\det(\mathbf{A}_2) = 4$, where $\mathbf{A}_2$ is the first principle minor of $\mathbf{A}_3$. The tridiagonal

matrix, $\mathbf{T}_3$, obtained by omitting the appropriate off-diagonal elements of $\mathbf{A}_3$, is indefinite. In fact $\det(\mathbf{T}_3) = -1$.

Now we try to establish some guidelines which will help us to identify classes of positive definite matrices with positive definite incomplete part. One such class is the class of diagonally dominant matrices. Setting any pair of off-diagonals symmetrically to zero will leave the matrix positive definite and symmetric. Another class is given by the symmetric H-matrices with positive diagonal (c.f. [11]).

Matrices arising from the finite element discretization of elliptic boundary value problems can be represented in the form

$$\mathbf{A} = \sum_{i=1}^{p} \mathbf{A}_i$$

where $\mathbf{A}_i$ represents the contribution of each finite element to the global matrix $\mathbf{A}$. The sum is over the total of p elements. $\mathbf{A}_i$ can be written more precisely as

$$\mathbf{A}_i = \mathbf{N}_i \, \mathbf{a}_i \, \mathbf{N}_i^T$$

where $\mathbf{a}_i$ is a small element stiffness matrix and $\mathbf{N}_i$ is an $n \times j$ boolean connectivity matrix with $j \ll n$. $\mathbf{N}_i$ are formal representations of the way in which the elements are joined together. In general, $\mathbf{a}_i$ is singular and $\mathbf{N}_i$ has full rank. For elliptic boundary value problems, $\mathbf{A}_i$ is a symmetric positive semi-definite matrix and by construction $\mathbf{A}$ must also be symmetric positive semi-definite. We can now state and prove the main proposition concerning the positive definiteness of incomplete $\mathbf{A}$.

Proposition: Let $\mathbf{A} = \sum_{i=1}^{p} \mathbf{A}_i$ be a symmetric positive definite matrix and let $\tilde{\mathbf{A}} = \sum_{\substack{i=1 \\ i \neq m}}^{p} \mathbf{A}_i + \tilde{\mathbf{A}}_m$, where $\tilde{\mathbf{A}}_m$ and $\mathbf{A}_i$, $i = 1, \cdots, p$, are all symmetric positive semi-definite. Then $\tilde{\mathbf{A}}$ is also positive definite if $N(\tilde{\mathbf{A}}_m) \subseteq N(\mathbf{A}_m)$, where $N(\mathbf{A}_i)$ denote the null space of $\mathbf{A}_i$.

Proof: $\mathbf{A}$ is positive definite iff $\bigcap_i N(\mathbf{A}_i)$ is trivial. Since $N(\tilde{\mathbf{A}}_m) \subseteq N(\mathbf{A}_m)$ then $\bigcap_{\substack{i=1 \\ i \neq m}}^{p} N(\mathbf{A}_i) \bigcap N(\tilde{\mathbf{A}}_m)$ is also trivial and the result follows.

This suggest that some of the off-diagonal elements of $\mathbf{A}_m$ can be removed symmetrically to get $\tilde{\mathbf{A}}$, without affecting the positive definiteness of the assembled matrix provided that the null space of $\tilde{\mathbf{A}}_m$ is contained in the null space of $\mathbf{A}_m$ and $\tilde{\mathbf{A}}_m$ is positive semi-definite. This result may seem rather cumbersome. However, as the following simple application demonstrates, it is very useful.

Consider the structure example of figure 1a. Each element consists of two nodes with two unknowns per node; the horizontal and vertical displacements. The resulting system of equations has a total of 10 unknows (because the total number of nodes is 7 with 2 on the boundry resulting in only 5 active nodes). Each element makes a contribution to the global matrix which couples the unknowns at the two nodes of the element. The explicit form of the element matrices are

$$\mathbf{a}_i = k_i \begin{bmatrix} c_i^2 & -c_i s_i & -c_i^2 & c_i s_i \\ -c_i s_i & s_i^2 & c_i s_i & -s_i^2 \\ -c_i^2 & c_i s_i & c_i^2 & -c_i s_i \\ c_i s_i & -s_i^2 & -c_i s_i & s_i^2 \end{bmatrix} = k_i \left[\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \otimes \begin{Bmatrix} c_i \\ s_i \end{Bmatrix} \begin{pmatrix} c_i & -s_i \end{pmatrix} \right]$$

where $c = \cos\theta_i$, $s = \sin\theta_i$; θ_i defines the oriantation of the i-th element and k_i is a positive quantity depending on the stiffness and the length of the each element. The coupling between the unknows at each node is eliminated when $\mathbf{a}_i$ is replaced by

$$\tilde{\mathbf{a}}_i = k_i \begin{bmatrix} c_i^2 & -c_i s_i & & \\ -c_i s_i & s_i^2 & & \\ & & c_i^2 & -c_i s_i \\ & & -c_i s_i & s_i^2 \end{bmatrix} = k_i \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{Bmatrix} c_i \\ s_i \end{Bmatrix} \begin{pmatrix} c_i & -s_i \end{pmatrix} \right]$$

$\tilde{\mathbf{a}}_i$ is positive semi-definite and $N(\mathbf{a}_i) \subseteq N(\tilde{\mathbf{a}}_i)$. This can be verified easily since the null vectors of $\tilde{\mathbf{a}}_i$ are $\mathbf{q}_1 = (\, s_i \quad c_i \quad 0 \quad 0 \,)^T$ and $\mathbf{q}_1 = (\, 0 \quad 0 \quad s_i \quad c_i \,)^T$ which are also the null vectors of $\mathbf{a}_i$. Therefore, by the above proposition, replacing any $\mathbf{a}_i$ by $\tilde{\mathbf{a}}_i$ will retain the positive definiteness of the global matrix. For example when this is done to the elements connected to nodes 1 and 5 the coupling between these nodes is eliminated as shown in Figure 1a.

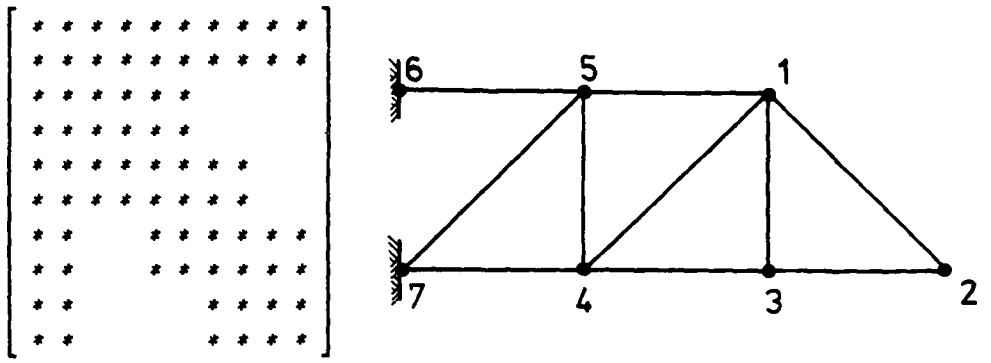


Figure 1a Original structure and its associated matrix.

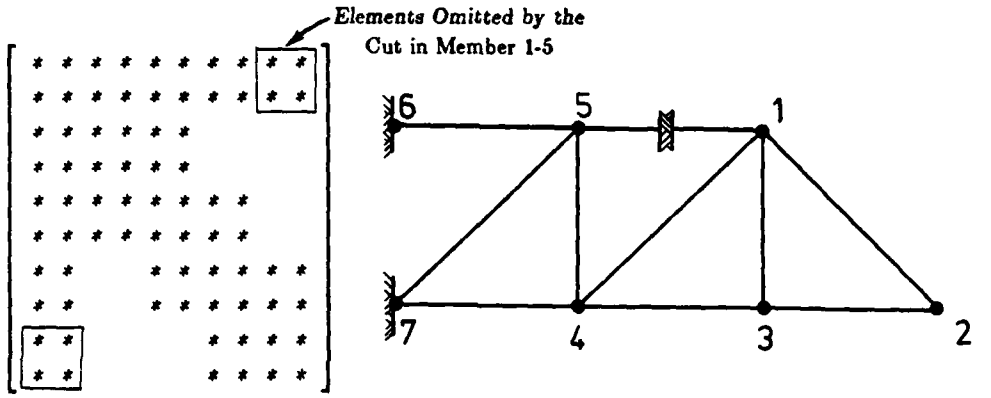


Figure 1b Tie-down structure and its associated matrix.

4. Implementation

Let $\mathbf{A} = \mathbf{M} + \mathbf{R}$, $\mathbf{M}$ positive definite, and $\mathbf{M} = \mathbf{C}\mathbf{C}^T$ the Choleski factorization of $\mathbf{M}$. Then banded preconditioning is best implemented by applying the standard CG method to the preconditioned problem

$$(\mathbf{I} + \mathbf{C}^{-1}\mathbf{R}\mathbf{C}^{-T})\mathbf{y} = \mathbf{C}^{-1}\mathbf{b} \tag{4}$$

with $\mathbf{y} = \mathbf{C}^T\mathbf{x}$, instead of using the algorithm of section 2. There are two good reasons for this. First if $\mathbf{M}$ is dense, i.e., no fill-in results when $\mathbf{M}$ is factored, then the Choleski factor $\mathbf{C}$ can be written over the storage locations provided for $\mathbf{M}$. In this case, the preconditioned algorithm does not need any extra storage. In contrast, incomplete factorization preconditioners [15] requires at least *NZA* or more extra storage cells in order to hold the incomplete factors. Here *NZA* denotes

the number of zeros in the upper triangular part of **A**. However, we should add that a certain class of incomplete factorization preconditioners, as in [7], require less storage than the more general versions. The second advantage of (4) is that the matrix vector product $(\mathbf{I} + \mathbf{C}^{-1}\mathbf{R}\mathbf{C}^{-T})\mathbf{y}$ needs exactly $2NZA$ multiply-adds, when **M** is dense. This only is n operations more than that needed in forming **Ay**. Hence, if **M** does not contain any zeros within the band, the banded preconditioning is not more expensive per step than the standard CG algorithm. When **M** does contain some zeros, the cost may be more but marginally.

There is an alternative way of utilizing the choleski factors of **M** for preconditioning. Note that

$$\mathbf{A} = \mathbf{C}(\mathbf{I} + \mathbf{C}^{-1}\mathbf{R}\mathbf{C}^{-T})\mathbf{C}^{T} \tag{5}$$

Therefore

$$\mathbf{A}^{-1} = \mathbf{C}^{-T}(\mathbf{I} + \mathbf{C}^{-1}\mathbf{R}\mathbf{C}^{-T})^{-1}\mathbf{C}^{-1} \tag{6}$$

Using the truncated Neumann series approach [5], we can find an approximation **H** to $\mathbf{A}^{-1}$ by

$$\mathbf{H} = \mathbf{C}^{-T}(\mathbf{I} - \mathbf{C}^{-1}\mathbf{R}\mathbf{C}^{-T})\mathbf{C}^{-1} \tag{7}$$

H can be used in connection with the PCG algorithm if (II) is replaced by

$$\mathbf{s}_0 = \mathbf{H}\mathbf{r}_0 \tag{8}$$

and (III.d) is replaced by

$$\mathbf{s}_{i+1} = \mathbf{H}\mathbf{r}_{i+1} \tag{9}$$

The convergence behavior of the two different implementations depends on the spectrum of **A**. Denote $\mathbf{C}^{-1}\mathbf{R}\mathbf{C}^{-T}$ by **W**. Then behavior of (4) is determined by the eigenvalues of $\mathbf{I} + \mathbf{W}$. Let λ be any eigenvalue of **W**; $\mathbf{I} + \mathbf{W}$ is positive definite, since **A** and **M** are both positive definite, then $1 + \lambda > 0$. Hence λ is greater than -1. Further, if the stronger condition $-1 < \lambda < 1$ holds, then the Neumann series can be applyed. The convergence behavior of PCG with the matrix of (7) as the preconditioner is governed by the eigenvalues of $\mathbf{HA} = \mathbf{C}^{-T}(\mathbf{I} - \mathbf{W})(\mathbf{I} + \mathbf{W})\mathbf{C}^{T}$. This has the same eigenvalues as $(\mathbf{I} - \mathbf{W})(\mathbf{I} + \mathbf{W}) = (\mathbf{I} - \mathbf{W}^2)$, which are $1 - \lambda^2$. Hence, $\kappa(\mathbf{I} + \mathbf{W}) = \dfrac{1 + \lambda_{max}}{1 + \lambda_{min}}$, where λ_{max} (respectively λ_{min}) is the largest

(smallest) eigenvalue of $\mathbf{W}$, and $\kappa(\mathbf{I} - \mathbf{W}^2) = \frac{1 - \lambda_0^2}{1 - \lambda_1^2}$, where λ_0 (respectively λ_1) is the eigenvalue closest to 0 (closest to 1). Hence one can not determine *a priori*, which of the the above implementations has better convergence properties. This choice depends on the problem.

The preconditioner matrix of (7) can be implemented by applying the CG method directly to

$$(\mathbf{I} - \mathbf{W}^2)\mathbf{y} = (\mathbf{I} - \mathbf{W})\mathbf{C}^{-1}\mathbf{b} \tag{10}$$

where $\mathbf{y} = \mathbf{C}^T\mathbf{x}$. This version nearly doubles the cost of each CG step, but may be less prone to orthogonality loss. However, as stated above all the eigenvalues of $\mathbf{W}$ must have a magnitude smaller than unity for the matrix of (10) to be positive definite. Unless the problem is better suited for the second implementation, the approach with (4) is preferable.

Finally, we would like to note that the algorithm described here can be extended to the case when $\mathbf{M}$ is not positive definite. In this case, $\mathbf{M}$ can be factored into $\mathbf{LDL}^T$ with $\mathbf{D}$ having both positive and negative terms. A preconditioning matrix of the form $\mathbf{L}\tilde{\mathbf{D}}\mathbf{L}^T$ can then be constructed, where the terms in $\tilde{\mathbf{D}}$ are the absolute values of the corresponding terms in $\mathbf{D}$. The preconditioned problem then becomes

$$(\mathbf{J} + \tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{L}^{-1}\mathbf{R}\mathbf{L}^{-T}\tilde{\mathbf{D}}^{-\frac{1}{2}})\mathbf{y} = \mathbf{C}^{-1}\mathbf{b} \tag{11}$$

where now $\mathbf{y} = \tilde{\mathbf{D}}\ \mathbf{L}^T\mathbf{x}$, and $\mathbf{J}$ is a diagonal matrix with elements 1 or -1.

5. Numerical Examples

The PCG with various forms of banded preconditioning was used in solving several large sparse symmetric systems of linear equations. One of our examples arises from finite element discretization of a rectangular plate. The other two examples are derived from finite difference approximations of elliptic partial differential equations. The examples and characteristics of each problem are described in more detail in tables 1 and 2. All matrices are included in the collection of "Sparse Matrix Test Problems" [6].

Example No.	Problem description
1	Poisson's equation in an L-shape region with mixed boundary conditions. The same problem as example 4 in [9], with less grid points.
2	$-\nabla(a\nabla u) = f$ in the unit cube in R^3 with Drichlet boundary conditions. a varies from 10^{-2} to 10^{-6}. Finite difference discretization with 7-point difference star and $9\times9\times9$ grid points is used with a random right-hand side.
3	Rectangular plate problem with one side fixed and the others free. A unit point load is applied at one of the free corners.

Table 1. Description of the Examples

Example	No. of Equations	No. of Nonzeros	Half Bandwidth	Fill-in	Condition Number
1	675	1965	30	9929	7.1×10^6
2	729	2673	81	40961	1.8×10^9
3	960	8402	43	66612	3.5×10^3

Table 2. Properties of the Test Matrices.

In the second column of Table 2., we list the number of nonzeros in the upper triangular part of the matrices, and in the third column the half band width; no attempt was made to reduce the band width by reordering the matrix. In the fourth column, we list the number of nonzeros in the triangular factorization after reordering the original matrix by the minimum degree algorithm.

In our numerical tests, we extracted the banded part of the test matrices for various band widths, m. The Choleski factors were computed using the LINPACK [4] subroutine DPBCO.

We applied the two implementations of banded preconditioning to our test problems. We compared our algorithms with the basic CG algorithm with diagonal scaling and with subroutine MA31 from the Harwell Library [15]. MA31 uses an incomplete factorization of the matrix as preconditioning with a drop tolerance, c. we used $c = -1.0$ and $c = 0.1$; the first choice allows no fill-in in the factorization where as the second permits all fill-in whose magnitude is greater than 0.1 relative to the diagonal elements. The tolerance may change in the routine depending on the amount of storage available to the program.

The results of the tests are given in the tables below. For each example, the tables list the half-band width m and the choice of c for MA31, the number of CG steps, the time in seconds for the two phases of the algorithm and the total time. In these tables, Methods 1 and 2 refer respectively to the simple banded preconditioning and banded preconditioning with the truncated Neumann series. All tests were carried out in double precision on a UNIVAC 1100. Our termination criterion was a reduction of the residual norm to a factor of 10^{-6} of its initial value.

Method	m , c	CG steps	Time for factor	Time for CG iter.	Total time
Basic CG	-	17	-	1.30	1.30
Method 1	1 2	15 15	0.31 0.36	1.38 1.45	1.69 1.81
Method 2	1 2	8 8	0.31 0.36	1.36 1.43	1.67 1.79
MA31	-1.0 0.1	2 1	0.44 0.46	0.22 0.15	0.66 0.61

Table 3 Results for Example 1.

Method	m , c	CG steps	Time for factor	Time for CG iter.	Total time
Basic CG	-	94	-	10.77	10.77
Method 1	1	85	0.32	9.92	10.24
	2	85	0.38	10.51	10.89
	8	85	0.73	14.03	14.82
	9	63	0.98	10.64	11.62
Method 2	1	43	0.32	9.51	9.83
	2	43	0.38	10.11	10.49
	8	43	0.38	13.63	14.42
	9	32	0.98	10.62	11.60
MA31	-1.0	56	0.62	5.52	6.14
	0.1	25	0.64	2.55	3.19

Table 4. Results for Example 2.

Method	m , c	CG steps	Time for factor	Time for CG iter.	Total time
Basic CG	-	> 200	-	(27.37)	(27.37)
Method 1	1	> 200	0.43	(88.28)	(88.71)
	2	111	0.51	49.36	49.87
MA31	-1.0	> 200	2.52	(102.67)	(105.19)
	0.1	> 200	2.67	(105.20)	(107.87)

Table 5. Results for Example 3.

Several conclusions can be drawn from the above results:

(i) Choice of the bandwidth. Clearly the choice of the bandwidth has to depend on the physical structure of the underlying problem. For Example 2, sensible choices of m are $m = 1$, $m = 9$, $m = 27$, because of the underlying structure of the matrix. The new information included in the band then leads to a drop in the number of CG steps and possibly in the total cost. As it turns out in Example 2, $m = 9$ is already more expensive than $m = 1$. $m = 2$ can not yield an improvement over $m = 1$, since the number of nonzeros of the preconditioning is not increased. We would like to note that by reordering the matrix it is possible to produce better preconditioners.

Similar considerations holds for examples 1 and 3. In Example 1 the next reasonable choice would have been $m = 30$ and in example 3 $m = 40$. These were not tried since the experience from example 2 seems to indicate that large m ($m > 10$ for example 2) will not improve the time for CG iteration to warrant the increase cost of factorization, unless the matrix is reordered.

(ii) Comparison of the two implementations. In the examples considered here there appears to be no significant difference in the overall performance of the two methods described here. It turns out that the use of Neumann series results in a slightly faster solution time for these examples. However, because of the arguments given in section 4, the first implementation is preferred.

(iii) Banded PCG versus Incomplete Factorization. We can draw no definitive conclusion for this comparison. In the first two examples, Ma 31 was about twice as fast as banded PCG. However, in Example 3 MA31 was not able to find a solution within 200 steps, whereas the simple banded PCG solved this difficult problem remarkably well.

The numerical examples, especially example 3, show that the banded PCG is a competitive alternative to preconditionings based on incomplete factorizations. It has several advantages which are not clearly reflected in the above tables which should be stressed here:

- The storage demands is the same or only slightly more than the standard CG method.
- The cost of one CG iteration does not increase when using this preconditioner.
- The factorization of M can be performed using existing codes.
- Conceptual simplicity.

References

[1] O. Axelson, "Solution of Linear Systems of Equations: Iterative Methods," *Sparse Matrix Techniques*, edited by V. A. Barker, Lecture Notes in Mathematics No. 572, 1977.

[2] R. Chandra, *Conjugate Gradient Methods for Partial Differential Equatons*, Ph.D. Thesis, Yale University, 1978.

[3] P. Concus, G. Golub, and D. O'leary, "A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations," *Proc. of the Symposium on Sparse Matrix Computations*, Argonne National Laboratory, Academic Press, 1975.

[4] J. J. Dongarra, J. R. Bundy, C. B. Moler, and G. W. Stewart, *LINPACK Users' Guide*, SIAM, Philadelphia, 1979.

[5] P. F. Dubois, A Greenbaum, and G. Rodrigue, " Approximating the inverse of a Matrix for Use in Iterative Algorithms on Vector Processors," *Computing 22*, pp. 257-268, 1979.

[6] I. Duff, R. Grimes, J. Lewis, and W. Poole. " Sparce Matrix Test Problems," *SIGNUM Newsletter*, No. 22, p. 22, 1982.

[7] S. Eisenstat, "Efficient Implementation of a Class of Preconditioned Conjugate Gradient Methods," *SIAM J. Sci. Stat. Comp.*, Vol. 2, pp. 1-4, 1981.

[8] L. Hageman, and D. Young, *Applied Iterative Methods*, Academic Press, 1981.

[9] D. Kershaw, "Incomplete Cholesky-Conjugate Gradient Method for the iterative Solution of Systems of Linear Equations," *J. Comp. Physics*, Vol. 26, pp. 43-65, 1978.

[10] M. R. Li, B. Nour-Omid, and B. N. Parlett, "A Fast Solver Free of Fill-In for Finite Element Problems," *SIAM J. Numer. Anal.* Vol. 19, pp. 1233-1242, 1982.

[11] T. Manteuffel, "An Incomplete Factorization Technique for Positive Definite Linear Systems," *Math. of Comp.*, Vol. 34, pp. 473-497, 1980.

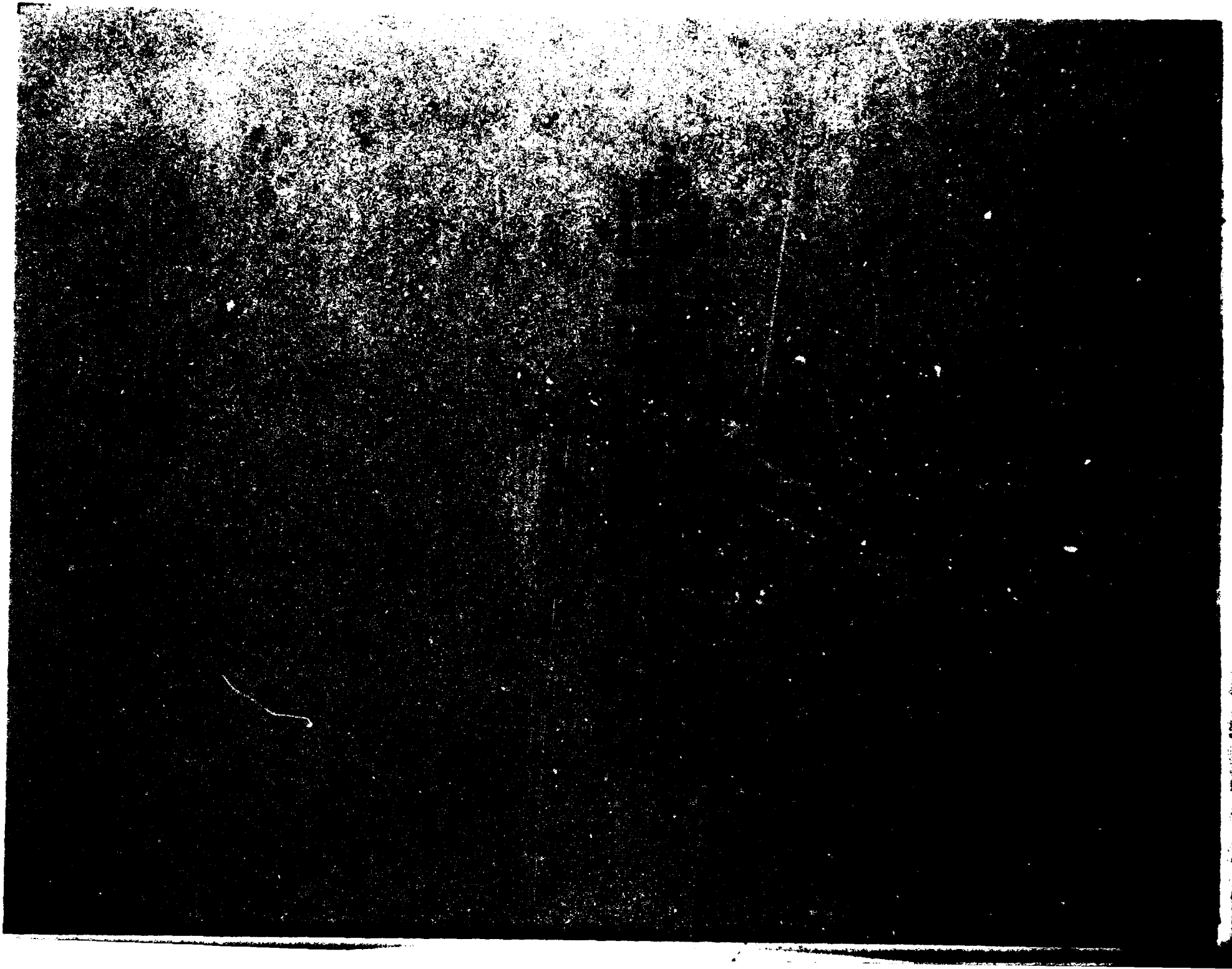
[12] J. Meijerink, and H. van der Vorst, "Guidelines for the Usage of Incomplete Decompositions in Solving Sets of Linear Equations as They Occur in Practical Problems", *J. Comp. Physics*, Vol. 44, pp. 134-155, 1981.

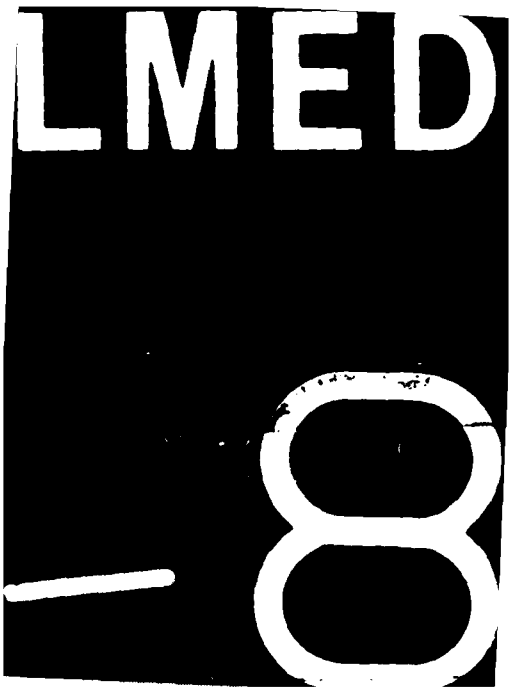
[13] J. Meijerink, and H. van der Vorst, "An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix", *Math. of Comp.*, Vol. 31, pp. 148-162, 1977.

[14] T. Miyakoda, *Reports 4C-6 and 2L-6*, Department of Applied Physics, Osaka University, Suita, Osaka 565, Japan, 1982.

[15] N. Munksgaard, "Solving Sparse Symmetric Sets of Linear Equations by Preconditioned Conjugate Gradient," *ACM Trans. on Math. Software*, Vol. 6, pp. 206-219, 1980.

[16] J. Reid, "On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations," *Large Sparse Sets of Linear Equations*, edited by J. Reid, Academic Press, pp. 231-254, 1971.




LMED
8